Лекция 1

1.1 Общие понятия о дискретных устройствах

Отрасль науки и техники об автоматически действующих устройствах и системах носит название ***автоматики***. Управление объектом и контроль его работы осуществляются в пределах сравнительно небольших расстояний. Для выполнения тех же функций на больших расстояниях, требующих для их преодоления специальных средств, применяют устройства ***телемеханики***.

Телемеханика – отрасль науки и техники, охватывающая теорию и технические средства контроля и управления объектами на расстоянии с применением специальных преобразователей сигналов для эффективного использования каналов связи.

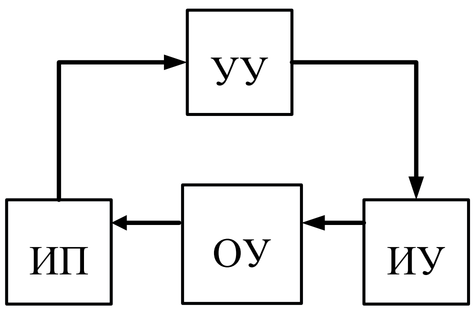

Рис. 1.1. Структурная схема системы автоматики

В ***автоматической*** системе (рис. 1.1) все функции управления производственным процессом, являющимся объектом управления (ОУ), осуществляются без участия человека. Управление каким-либо объектом – это воздействие на него с целью обеспечения требуемого течения процессов в объекте или заданного изменения его состояния. Основой управления является получение и обработка информации о состоянии объекта и внешних условиях его работы для определения воздействий, которые необходимо приложить к объекту, чтобы достичь цели управления.

От ОУ, например, технологического процесса или какого-либо устройства, поступает ***осведомительная информация***, характеризующая его состояние. Для сбора осведомительной информации применяют специальные измерительные приборы (ИП): чувствительные элементы, датчики, измерительные устройства, преобразователи различных типов и т. п. Эта информация поступает в управляющее устройство

(УУ). В результате обработки полученной информации УУ, в соответствие с алгоритмом управления, выясняет характер требуемых управляющих воздействий на ОУ.

УУ выдает управляющие воздействия на исполнительные устройства (ИУ), которые и воздействуют на ОУ. Полученные воздействия изменяют состояние ОУ, которое вновь контролируется ИП. Таким образом, этот процесс повторяется постоянно во времени.

1.2 Электрические сигналы

Для передачи и переработки информации ее представляют в некоторой форме с использованием различных знаков. В общем случае под информацией понимают совокупность сведений о событиях, объектах или явлениях. Совокупность знаков, содержащих ту или иную информацию, называют сообщением. Так, при телеграфной передаче сообщением является текст телеграммы, представляющий собой последовательность отдельных знаков – букв и цифр. Сообщение может иметь самое различное содержание, но независимо от этого всегда отображается в виде ***сигнала***.

В качестве сигнала можно использовать любой физический процесс, изменяющийся в соответствии с переносимым сообщением. Мы будем рассматривать только электрические сигналы, физической величиной, которых, является ток или напряжение. Сигналы формируются изменением (***модуляцией***) тех или иных параметров амплитуды, фазы, частоты. Сигнал – это средство перенесения информации в пространстве и времени. Для соответствия между сообщением и сигналом, т.е. для обеспечения возможности извлечения сообщения из полученного сигнала, последний следует формировать по определенным правилам. Каждому сообщению должен соответствовать свой сигнал. Построение сигнала по определенным правилам называют ***кодированием***.

Если сигнал (или сообщение) может принимать любые значения в некотором интервале времени, его называют ***непрерывным***, или ***аналоговым***. Такой сигнал является непрерывной функцией от времени (на каком-то интервале), даже если сообщение к таковой не относится (рис. 1.2).

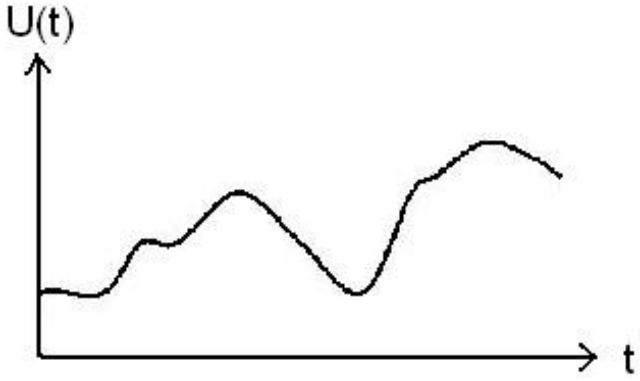

Рис. 1.2. Аналоговый сигнал

Если сигнал (или сообщение) принимает только некоторые определенные значения из некоторого множества, то такой сигнал называют ***дискретным*** (рис. 1.3). Здесь на одинаковых промежутках времени (Δt), называемых дискретами времени, значение сигнала не изменяется.

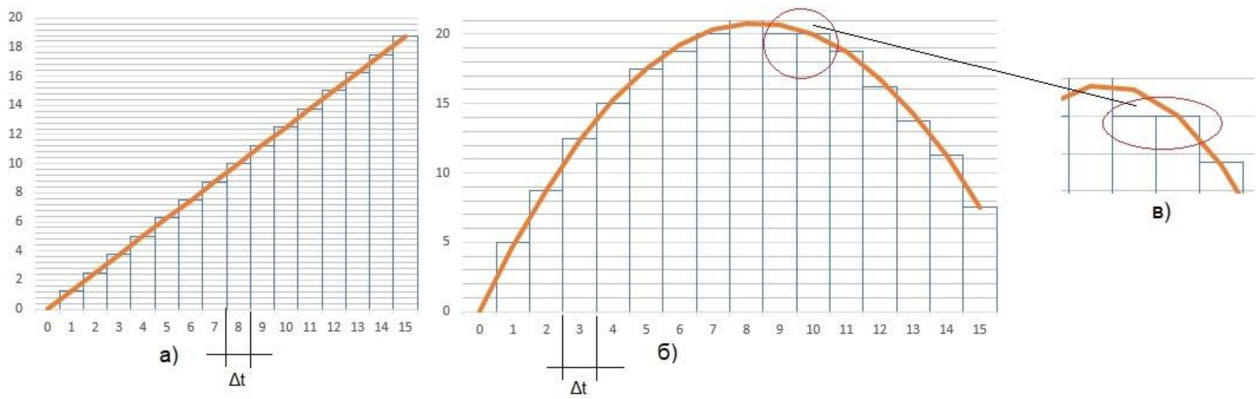

Рис. 1.3. Дискретный сигнал

Диапазон изменения измеряемой величины ΔU делится на число N, показывающее какое количество дискретных «ступенек» будет участвовать в процессе формирования дискретного сигнала, подобного исходному. Число представляется в двоичном коде $N=2^n$. Тогда считается, что n – это разрядность двоичного числа. Каждому двоичному коду числа соответствует свое напряжение U. На рис. 1.3а показан линейно нарастающий аналоговый сигнал и соответствующие ему ступеньки напряжения дискретного сигнала. На рис.1.3б показаны нелинейный аналоговый и дискретный сигналы. Очевидно, что ступенчатый дискретный сигнал может соответствовать аналоговому с погрешностью, что и видно в точках 9 и 10 (рис. 1.3в).

Если дискретный сигнал представляет собой функцию, которая может принимать только два значения (одно из них обычно обозначают 1, а другое 0), то его называют ***логическим*** или ***цифровым*** сигналом (рис. 1.4). Устройства, обрабатывающие такие сигналы, называют ***цифровыми*** устройствами.

Логические сигналы широко используются при построении устройств автоматики. При этом подразумевается независимость в обозначении сигнала от истинного значения действующей амплитуды напряжения.

Так, для некоторых микросхем, цифровой сигнал может принимать значение высокого уровня, равное +15 В, для других +5 В. И в том и в другом случае говорят, что сигнал принимает единичное значение, и обозначают его 1. Противоположный сигнал, по уровню близкий к нулевому значению, обозначают 0 и называют нулевым.

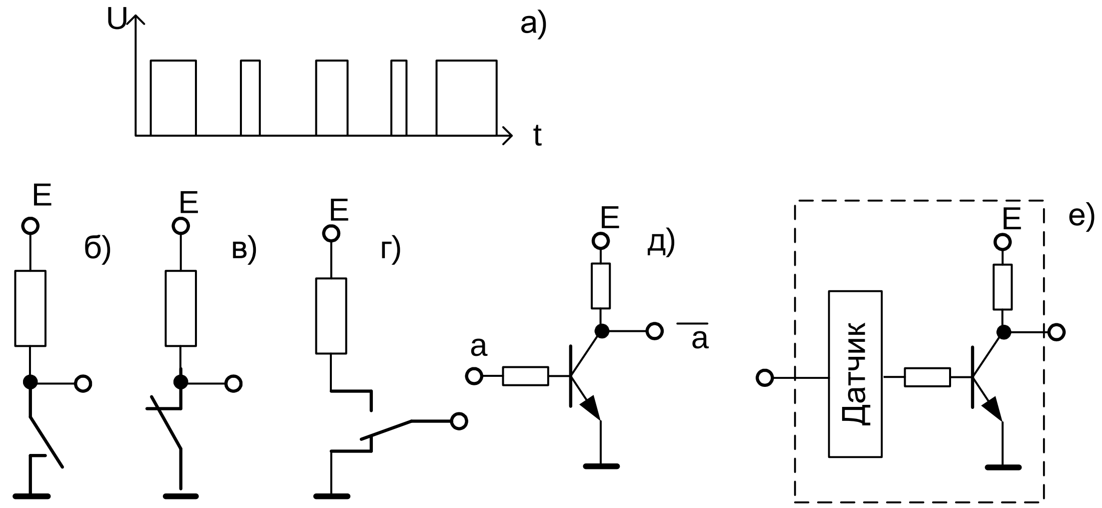

Рис. 1.4. Цифровой сигнал и его формирование
а – нормально разомкнутый (НР) контакт;
б – нормально замкнутый (НЗ) контакт;
в – переключающий (ПК) контакт;
г – электронный ключ; е – электронный датчик

Как может быть сформирован цифровой сигнал. Могут быть использованы контактные или электронные элементы. На рис. 1.4 а показана схема с контактом, который в неактивном состоянии разомкнут. На выходном полюсе будет присутствовать высокий уровень, равный Е питания, т.е. логическая единица. Если на контакт нажать, то на выход будет подключена «земля» - логический ноль. При прекращении нажатия на контакт он самостоятельно возвращается в неактивное состояние. На рис.1.4 б показана схема, формирующая логические сигналы с помощью нормально замкнутого контакта, который в неактивном состоянии подключает на выход 0, а при нажатии на него выдает 1. На рис. 1.4 в показан переключающий контакт, который в любой момент времени соединяется с одним или другим контактом и остается в таком положении, пока не переключить его. На рис. 1.4 г показан электронный ключ, который на вход (в базу) принимает электрический сигнал и на выходе (коллектор) формирует противоположный сигнал, т.е. реализует операцию инвертирования. Во многих современных устройствах используется электронный датчик (рис. 1.4е). В его состав входит датчик, который преобразует входную измеряемую величину в электрический сигнал. Он поступает на вход транзисторного ключа, который и выдает логический сигнал. Этот сигнал показывает достигла ли измеряемая величина заданного значения или нет. Например, датчик давления, настроенный на какую-то величину давления пара в котле может показать достижение давления пара какой то критической величине формированием уровня 0 или 1 на выходе.

Все устройства (рис. 1.4 а,б,в,г,е) могут входить в состав разнообразных технических устройств. Они могут переключаться при воздействии механической силы, или воздействия магнитного поля, или других физических явлений.

Таким образом цифровые или логические устройства имеют дело с логическими и дискретными сигналами в числовой форме. Для обработки чисел необходимо знать правила их представления и арифметические операции над ними. Это определяется системами счисления, т.е. законом записи чисел.

1.3. Системы счисления

Под системой счисления будем понимать способ записи чисел с помощью цифровых знаков. В настоящее время используются так называемые позиционные системы счисления, в которых изменения положения цифры в записи числа ведет к изменению самого числа, что не всегда выполняется в непозиционных системах счисления. Позиционной системой счисления (СС) называется такая, которая удовлетворяет следующему равенству:

$$A_{(q)} = a_n * q^n + a_{n-1} * q^{n-1} + \cdots + a_1 * q^1 + a_0 * q^0 + a_{-1} * q^{-1} + \cdots + a_{-m} * q^{-m} \qquad (1.1)$$

где

$A_{(q)}$ – запись числа в q-ичной СС;

q – основание СС (целое число, лежащее в диапазоне от 1 до N);
$a_к$ – значение к-го разряда в записи числа (a=0, 1, …, q-1);
n – количество целых разрядов в записи числа;
m – количество дробных разрядов в записи числа, конкретные значения чисел *n* и *m* будем называть номером разряда или разрядом числа;
q^P – будем называть весом разряда (P = n, n-1, …, 1, 0, -1, -2, …, -m).

Так, для десятичной (десятеричной) СС q = 10, $a_к$ = 0, 1, 2, …9, поэтому числа записываются следующим образом:

$$243{,}98_{(10)} = 2 \cdot 10^2 + 4 \cdot 10^1 + 3 \cdot 10^0 + 9 \cdot 10^{-1} + 8 \cdot 10^{-2} = 200 + 40 + 3 + 0{,}9 + 0{,}08 .$$

Изменяя q можно получать запись одного и того же числа в разных СС. Существуют разные алгоритмы перевода чисел из одной СС в другую. Рассмотрим один из них (*рассматриваются СС, которые используются в цифровых устройствах – двоичная, восьмеричная и шестнадцатеричная*).

Перевод числа в другую СС производится ***отдельно для целой и дробной*** части числа. Целая часть переводится делением, а дробная умножением на основание новой СС.

Перевод целой части:

1) число, которое нужно перевести из одной СС в другую, делится нацело на основание новой СС (q) с записью остатка от деления (остаток не может быть больше q-1);
2) полученное частное снова делится нацело на q;
3) пункт 2 выполняется до тех пор, пока полученное частное не окажется меньше основания новой СС;
4) для получения записи числа в новой СС к последнему частному приписываются справа в обратном порядке все остатки от деления, полученные при выполнении пунктов 1, 2, 3. Например, десятичное число $167{,}36_{(10)}$ так переводится в двоичную СС (рис. 1.5):

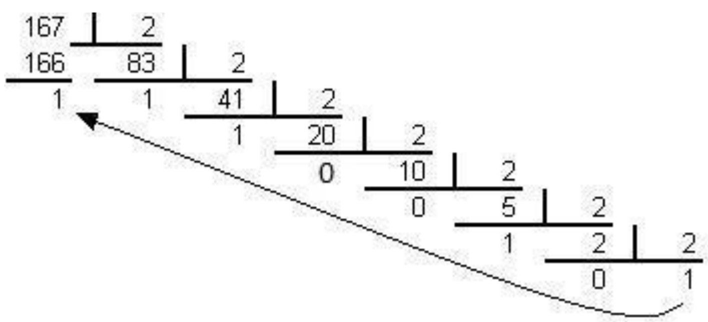

Рис. 1.5 Перевод десятичного числа в двоичную СС

Результат перевода: $167_{(10)}=10100111_{(2)}$.

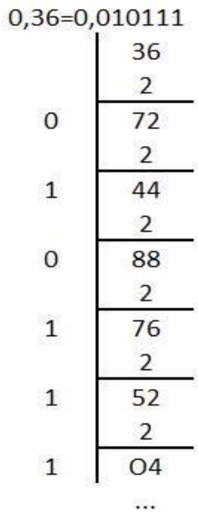

Рис. 1.6 Перевод десятичного числа в двоичную СС

Дробная часть числа:

1. дробная часть умножается на основание новой СС;
2. полученная целая часть отделяется от произведения;
3. пункты 1 и 2 продолжаются до тех пор, пока полученной произведение не окажется равным нулю, или до достижения заранее оговоренного количества разрядов;
4. дробная часть в новой СС получается из последовательности получаемых целых частей произведений в порядке их получения:

К целой части числа приписывается дробная часть, и тогда

$$167{,}36_{(10)}=10100111{,}010111_{(2)}.$$

Для обратного перевода из двоичной СС в десятичную можно использовать формулу 1.1:

$$10100111{,}010111_{(2)} =$$
$$=(\textit{целое})1*2^7+0*2^6+1*2^5+0*2^4+0*2^3+1*2^2+1*2^1+1*2^0+$$
$$+(\textit{дробное})+0*2^{-1}+1*2^{-2}+0*2^{-3}+1*2^{-4}+1*2^{-5}+1*2^{-6}=$$
$$=128+32+4+2+1+0{,}25+0{,}0625+0{,}03125+0{,}015625=167{,}3594$$

Как видно из полученного результата целая часть переводится точно, а дробная часть отличается от дробной части исходного десятичного числа. Если обратить внимание на формулу 1.1, то очевидно, что дробная часть может приближаться к истинному значению только при увеличении количества разрядов (в пределе к бесконечному). Поэтому при переводах из одной СС в другую необходимо определить такое количество дробных разрядов, которое с достаточной степенью точности представляют исходные числа. Так в современных компьютерах дробная часть представляется в виде 32-х или 64-х двоичных разрядов.

Для чего нужна двоичная СС? При проектировании первых цифровых электронных вычислительных машин были сформулированы принципы, которые получили название принципов фон Неймана. В соответствии с ними принято использование в ЦЭВМ двоичной СС.

С другой стороны, очевидно, что для записи одного и того же числа в разных СС потребуется разное количество разрядов. Число 167,36 десятичной СС записано пятью разрядами. В то же время для двоичной СС потребуется 11 и даже больше разрядов. т. е. чем больше основание СС, тем меньшим числом разрядов может быть записано одно и тоже число.

В цифровых устройствах кроме двоичной СС применяются восьмеричная и шестнадцатеричная СС. Их основное назначение – это отображение состояния двоичных устройств более коротким числом разрядов.

Восьмеричная СС оперирует восемью цифрами 0,1,2,3,4,5,6,7. Перевод из десятичной СС в восьмеричную:

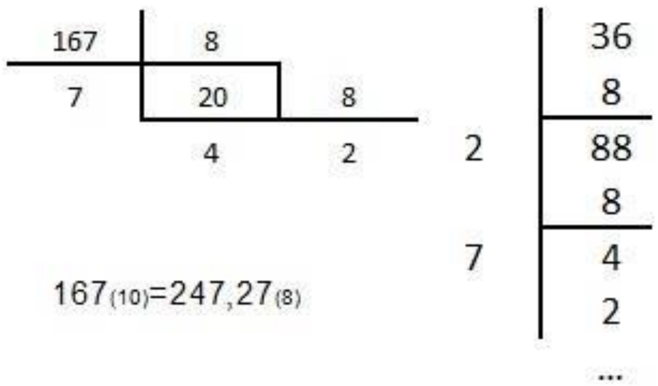

Рис. 1.7. Перевод десятичного числа в восьмеричную СС

Шестнадцатеричная СС должна использовать 16 цифр, из которых десять – это десятичные цифры 0,1,2,3,4,5,6,7,8,9, а шесть цифр обозначаются символами латинского алфавита A,B,C,D,E,F (a,b,c,d,e,f). При этом остатки от делений и последнее частное полученные при переводе целой части и результаты целой части, полученные при умножении дробной части заменяются на буквенные символы следующим образом:

10 – A, 11 – B, 12 – C, 13 – D, 14 – E, 15 – F. Например, перевод 167,36 в шестнадцатеричную СС:

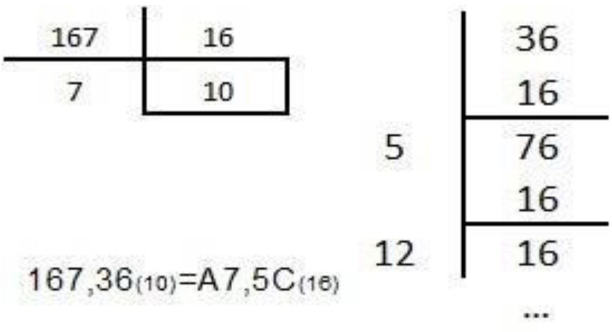

Рис. 1.8 Перевод десятичного числа в шестнадцатеричную СС

Примеры соответствия чисел в различных СС приведены в табл.1.1, из которой видно, что чем больше величина числа, те меньше разрядов нужно для СС с большим основанием:

Таблица 1.1

Десятичная	Восьмеричная	Шестнадцатеричная	Двоичная
28	34	1C	11100
493	755	1ED	111101101
16983	41127	4257	100001001010111

При обратном переводе из двоичной СС в десятичную нужно в формулу 1.1 подставить необходимые значения и вычислить полученную сумму. Например:

$$100101{,}101_2 = 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2}$$
$$+1\cdot 2^{-3} = 16 + 2 + 0{,}5 + 0{,}125 = 18{,}625_{10}$$

$$C9,F1_{16} = C * 16^1 + 9 * 16^0 + F * 16^{-1} + 1 * 16^{-2}$$
$$= 12 * 16^1 + 9 * 16^0 + 1 * 16^{-1} + 1 * 16^{-2} =$$
$$= 192 + 9 + 0{,}9375 + 0{,}0039 = 201{,}94_{10}$$

$$175{,}45_8 = 1 * 8^2 + 7 * 8^1 + 5 * 8^0 + 4 * 8^{-1} + 5 * 8^{-2}$$
$$= 64 + 56 + 5 + 0{,}5 + 0{,}078 = 125{,}58_{10}.$$

Цифровые устройства, как было сказано ранее используют двоичную СС. Один разряд двоичного числа, имеющий значение 0 или 1, носит название **бит** (bit). Но это самая мелкая единица измерения информации. Цифровые и микропроцессорные устройства оперируют более крупными единицами. Восемь бит образуют 1 **байт**. Байты объединяются в **слова**, которые, в зависимости от разрядности устройства, состоят из 2, 4 или 8 байт.

1.4 Представление чисел в разрядной сетке цифровых устройств

Цифровые устройства предназначены для обработки информации, записанной в двоичной СС, или по-другому в двоичном коде. Над числами выполняются арифметические и логические операции. Числа могут быть целыми и дробными, положительными и отрицательными.

Числа размещаются в разрядной сетке (рис. 1.9) цифрового устройства и все операции выполняются поразрядно над разрядами с одним весом. Разряды нумеруются, начиная с младшего, который имеет номер ноль. Формат такого расположения:

7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	1

Рис. 1.9 Разрядная сетка 8-ми разрядного цифрового устройства

Числа, с которыми оперирует цифровое устройство, могут быть беззнаковыми целыми (рис. 1.9)., т. е. все разряды являются значащими. Диапазон чисел, размещаемый в такой разрядной сетке, изменяется от 0 до 2^8-1=255. В представленном рисунке записано число $145_{(10)}$.

Если числа являются целыми со знаком, то знак размещается в старшем разряде, причем знак положительных чисел кодируется нулем, отрицательных – единицей. 1,0010001 – отрицательное число, 0,0010001 – положительное число. На рис. 1.10 показано размещение чисел в разрядной сетке устройства. Знаковый разряд отделен двойной чертой.

а)

7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	1

б)

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1

Рис. 1.10. Разрядная сетка цифрового устройства со знаковым разрядом

а – отрицательное число -17;

б – положительное число +17

Диапазон чисел в этом случае изменяется. Половина диапазона из 256 чисел отдается под положительные числа, половина под отрицательные.

Если в этой разрядной сетке размещаются числа со знаком и с дробной часть, заранее оговаривается сколько знаков отдается под целую, а сколько под дробную часть (рис. 1.11). Т.о. в таком представлении появляются две запятые. Одна – знаковая, другая – десятичная. Такое число записывается так: 1,1001,101 для отрицательного числа, 0,1011,011 – для положительного.

Рис. 1.11. Разрядная сетка цифрового устройства с целой и дробной частью

Очевидно, что такое количество дробных разрядов в очень невелико для представления чисел с дробной частью. Поэтому такое разделение практически не применяется. Чтобы получить более точное значение дробной части, создается двух байтовое слово, в котором целая часть – это старший байт со знаковым разрядом. Младший байт содержит только дробную часть числа и это целое беззнаковое число (рис. 1.12).

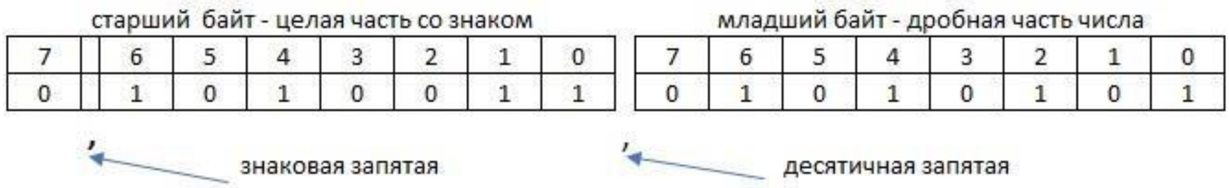

Рис. 1.12. Двух байтовое число с дробной частью

При необходимости создаются 4-х или даже 8-ми байтовые слова, в которых большая часть отдается под дробную часть. В таких случаях все обрабатываемые числа представляются в одном формате, т. е. во всех числах десятичная запятая располагается в **строго определенном месте** между разрядами. Такой формат чисел носит название – числа с **фиксированной запятой** – $0{,}1010011{,}01010101_2$. В современных ЭВМ все числа в формате с фиксированной запятой при вводе преобразуются так, чтобы оказались <1. В этом случае все разряды ЭВМ, кроме знакового, оказываются принадлежащими дробной части.

1.5 Арифметические операции в цифровых устройствах

В начале приведем таблицу (табл. 1.2), в которой приведено соответствие десятичных чисел, цифрам восьмеричной, шестнадцатеричной и двоичной СС.

Таблица 1.2

Десятичные числа	Двухразрядные числа	Трехразрядные двоичные числа	Четырехразрядные двоичные числа	Восьмеричные цифры	Шестнадцатеричные цифры
0	00	000	0000	0	0
1	01	001	0001	1	1
2	10	010	0010	2	2
3	11	011	0011	3	3
4		100	0100	4	4
5		101	0101	5	5
6		110	0110	6	6
7		111	0111	7	7
8			1000		8
9			1001		9
10			1010		A
11			1011		B
12			1100		C
13			1101		D
14			1110		E
15			1111		F

Выполнение арифметических операций в цифровых устройствах будем рассматривать на примере пяти разрядного устройства со знаковым разрядом (рис. 1.13). Десятичная запятая перенесена правее младшего разряда, т. е. в такой разрядной сетке могу размещаться только целые числа.

Рис. 1.13 Формат разрядной сетки цифрового устройства

На рисунке приведены веса каждого из разрядов для быстрого перевода чисел из двоичного в десятичное, также можно использовать и табл. 1.2.

Правила сложения двоичных кодов чисел:

- числа a и b подписываются одно под другим так, чтобы разряды с одним весом оказались одно под другим;
- сложение выполняется поразрядно по правилам, указанным в табл. 1.3;
- в операцию сложения вступают все разряды, включая знаковые;

Таблица 1.3

a_i	b_i	Перенос p_i	Сумма i-тых разрядов
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- перенос, возникающий в i-том разряде, прибавляется к сумме i+1 разряда.

1.5.1 Прямой код двоичного числа

Если к двоичному коду числа добавить знаковый разряд, то получим прямой код числа (ПК). Например:

0,1001 (ПК) = 1*8+0*4+0*2+1*1 = +10;

0,0011 (ПК)= 0*8+0*4+1*2+1*1 = +3;

1,1010 (ПК)= 1*8+0*4+1+2+0*1 = -10;

1,1110 (ПК)= 1*8+1*4+1*2+0*1 = -14.

Примеры сложения положительных чисел:

a	+10	0	,	1	0	1	0
b	+3	0	,	0	0	1	1
a+b	+13	0	,	1	1	0	1

+5	0	,	0	1	0	1
+6	0	,	0	1	1	0
+11	0	,	1	0	1	1

+11	0	,	1	0	1	1
+3	0	,	0	0	1	1
+14	0	,	1	1	1	0

Примеры сложения с участием отрицательных чисел:

a	+5	0	,	0	1	0	1
b	-2	1	,	0	0	1	0
a+b	**-7**	**1**		**0**	**1**	**1**	**1**

Все числа в ПК

-7	1	,	0	1	1	1
-6	1	,	0	1	1	0
+13	**0**	**,**	**1**	**1**	**0**	**1**

-1	1	,	0	0	0	1
-9	1	,	1	0	0	1
+10	**0**	**,**	**1**	**0**	**1**	**0**

Как видно, при сложении отрицательных чисел получены неверные результаты.

Отсюда следует:

<u>в прямом коде можно выполнять сложение только положительных чисел.</u>

1.5.2 Дополнительный код двоичного числа

Для правильного выполнения операции сложения с участием отрицательных чисел разработаны дополнительный и обратный код. В современных микропроцессорных устройствах используется, как правило, дополнительный код (ДК). ДК образуется по следующим правилам:

1. для **положительных** чисел **ДК и ПК совпадают**;
2. все разряды отрицательного числа (**<u>за исключением знакового</u>**) инвертируются и к младшему разряду прибавляется единица (*при инвертировании 0 в разряде заменяется на 1, а 1 заменяется на 0*);
3. единица переноса из знакового разряда теряется (*в микропроцессорах она запоминается в специальном устройстве – триггере переноса*);
4. сумма, полученная в результате сложения чисел, представленных в ДК, тоже является ДК;
5. если при сложении чисел получается отрицательное число, то чтобы узнать его **истинное** значение, его нужно перевести в ПК, причем для такого перевода следует применить правило из пункта 2. Примеры перевода из ПК в ДК:

-7	1,	0	1	1	1	ПК
	1,	1	0	0	0	инвертирование разрядов
					1	прибавление 1
	1,	1	0	0	1	ДК числа -7

-4	1,	0	1	0	0	ПК
	1,	1	0	1	1	инвертирование разрядов
					1	прибавление 1
	1,	1	1	0	0	ДК числа -4

При использовании ДК предыдущие примеры выполняются верно:

a	+5	0	,	0	1	0	1	
b	-2	1	,	1	1	1	0	ДК
a+b	+3	**0**		**0**	**0**	**1**	**1**	

Результат положительный ДК совпадает с ПК, перевод не нужен

-7	1	,	1	0	0	1	ДК
-6	1	,	1	0	1	0	ДК
	1	**,**	**0**	**0**	**1**	**1**	ДК

Перевод из ДК в ПК

	1,	1	1	0	0
					1
-13	1,	1	1	0	1 ПК

-1	1	,	1	1	1	1	ДК
-9	1	,	0	1	1	1	ДК
	1	**,**	**0**	**1**	**1**	**0**	

Перевод из ДК в ПК

	1,	1	0	0	1
					1
-10	1,	1	0	1	0 ПК

1.5.3 Переполнение разрядной сетки

В разрядной сетке цифрового устройства (рис. 1.13) могут быть представлены положительные числа от нуля до $+15_{10}$ (0,0000 до 0,1111) и отрицательные от -15_{10} до -1_{10} (1,0001 до 1,1111). Если при сложении чисел результат превышает +15 или -15, то говорят, что происходит арифметическое переполнение (АП) разрядной сетки цифрового устройства. Рассмотрим несколько примеров (рис. 1.14).

а)

a	+11	0	,	1	0	1	1	ПК
b	+8	0	,	1	0	0	0	ПК
a+b		1	,	0	1	1	1	ПК
	+19							

б)

-7	1	,	0	1	1	1	ПК
-13	1	,	1	1	0	1	ПК
перевод из ПК в ДК и сложение							
	1	,	1	0	0	1	ДК
	1	,	0	0	1	1	ДК
-20	0	,	0	1	0	0	ДК

в)

-9	1	,	1	0	0	1	ПК
-7	1	,	0	1	1	1	ПК
перевод из ПК в ДК и сложение							
	1	,	0	1	1	1	ДК
	1	,	1	0	0	1	ДК
-16	1	,	0	0	0	0	ДК

Рис. 1.14 Арифметическое переполнение разрядной сетки
а) переполнение при положительных числах;
б,в) переполнение при отрицательных числах;

Правила:

1. если при сложении ***положительных*** чисел в знаковом разряде ***вместо нуля образовалась единица***, то произошло АП. В примере (рис. 1.14 а) должна получиться сумма +19, что превышает максимальное число +15, которое может разместиться в разрядной сетке.
2. если при сложении ***отрицательных*** чисел в знаковом разряде ***вместо единицы образовалсяь ноль***, то произошло АП. В примере (рис. 1.14 б) должна получиться сумма -20, что превышает максимальное число -15, которое может разместиться в разрядной сетке.

В примере (рис. 1.14 в) слагаемые имеют отрицательный знак и знак результата тоже отрицателен. Но сумма, которая должна получиться равна -16, что также превышает возможности разрядной сетки (-15).

3. Признаком АП в этом случае является то, что ***знаки слагаемых и результата отрицательны и во всех значащих разрядах суммы получились нули.***

1.5.4 Арифметические операции вычитания, умножения, сложения.

Операция арифметического вычитания A-B может быть заменена на операцию сложения A-B = A+(-B), хотя и существуют алгоритмы вычитания. Вычитание i-тых разрядов чисел подчиняются правилу, по которому при вычитании единицы из нуля, производится заем из старшего i+1 разряда:

вычитание А-В

заем из a_{i+1}	a_i	b_i	Разница i-тых разрядов
0	0	0	0
1	0	1	1
0	1	0	1
0	1	1	0

Пример, вычитания двух чисел (без знака с дробной частью):

										,				
A	1	0	1	0	1	1	1	0	1	,	1	1	=	349,8
B	0	1	0	0	1	0	1	1	1	,	0	1	=	151,3
A-B	0	1	1	0	0	0	1	1	0	,	1	0	=	198,5

Операция умножения выполняется поразрядным умножением каждого разряда множителя на все разряды множимого, сохранением этого промежуточного результата и сложения всех промежуточных результатов:

множимое А							1	0	1	1	0	1	1	A=91
множитель В							0	0	1	1	0	0	1	B=25
промежуточные произведения каждого разряда							1	0	1	1	0	1	1	
множителя на множимое						0	0	0	0	0	0	0		
					0	0	0	0	0	0	0			
				1	0	1	1	0	1	1				
			1	0	1	1	0	1	1					
		0	0	0	0	0	0	0						
	0	0	0	0	0	0	0							произведение
сумма всех промежуточных произведений														A*B
	0	1	0	0	0	1	1	1	0	0	0	1	1	2275

При перемножении двух n-разрядных чисел получается 2*n разрядное число. Алгоритмы деления и умножения относятся в микропроцессорной технике к длинным операциям, выполняемым в течении нескольких машинных циклов. Они реализуются относительно сложными алгоритмами, требуют определенных аппаратных ресурсов и реализуются, как правило, программным способом.

1.6 Двоично-десятичные коды.

Существует еще одна система кодирования чисел, называемая двоично-десятичным. Ее еще называют кодированием 8-4-2-1. Т. е. каждая десятичная цифра 0, 1,...9 заменяется четырехразрядной ***двоичной - десятичной декадой*** (ДДД). Достоинством такого кодирования заключается в ее простоте и точном переводе из десятичной системы счисления. Например: 1972,68=0001 1001 0111 0010, 0110 1000.

Этот способ кодирования удобен тем, что в одном байте помещается две цифры ДДД. Недостатком является избыточность такого кода, т. к. он требует большего числа двоичных разрядов и сложность арифметических операций. Дело в том, что при сложении двух ДДД полученная в ней сумма может превысить цифру 9, которая является максимальной в этой системе кодирования. Поэтому ко всем декадам больше 9 прибавляется корректирующий код шестерки (0110). Если и после этого образуются декады большие 9, то они опять корректируются. Это может продолжаться несколько раз в зависимости от слагаемых. Поэтому сложение двоично – десятичных чисел, в среднем, по времени длится дольше, чем при сложении обычных двоичных чисел.

Например:

769,58	=	0111	0110	1001	,	0101	1000
154,12	=	0001	0101	0100	,	0001	0010
923,70	=	1000	1011	1101	,	0110	1010
			больше 9	больше 9			больше 9

	1000	1011	1101	,	0110	1010
корректирующий код		0110	0110			0110
Сумма	1001	0010	0011	,	0111	0000
	9	2	3	,	7	0

Сначала числа складываются, а затем выполняется коррекция. При сложении длинных чисел, например 8 байт, может потребоваться применение операции корректирования несколько раз.

2. Логические основы цифровых устройств. Булевы функции

Математическая модель цифрового дискретного устройства S представляет собой шестикомпонентный вектор

$$S = (A, Z, W, \delta, \lambda, a_1)$$

где $\mathbf{A=\{a_1,\dots a_m,\dots a_M\}}$ – множество внутренних состояний устройства;

$\mathbf{Z=\{z_1,\dots z_f,\dots z_F\}}$ – множество входных сигналов;

$\mathbf{W=\{w_1,\dots w_g,\dots w_G\}}$ – множество выходных сигналов;

$\mathbf{\delta: A \times Z \to A}$ – функция переходов, реализующая отображение произведения множеств $A \times Z$ на A;

$\mathbf{\lambda: A \times Z \to W}$ – функция выходов, реализующая отображение произведения множеств $A \times Z$ на W;

$\mathbf{a_1 \subseteq A}$ – начальное состояние устройства.

Частным случаем являются устройства, для которых множество A является пустым ($\mathbf{A=\varnothing}$). Тогда они описываются только множеством входных сигналов Z, выходных сигналов W и функцией выходов

$$\mathbf{\lambda: Z \to W},$$

которая устанавливает соответствие между элементами множества Z входных сигналов и множества W выходных сигналов. Такие устройства относятся к устройствам первого рода и носят название ***комбинационные схемы*** (КС). Для них характерно то, что выходной сигнал однозначно определяется только входным сигналом.

2.1 Основы булевой алгебры

В XIX веке английский математик Джордж Буль разработал основные положения алгебры логики, которую называют булевой алгеброй. Только в XX веке соответствующий уровень развития промышленности привел к тому, что этот раздел математики оказался востребованным.

Это произошло в связи с внедрением устройств автоматики, построенных на реле. Реле – это электромеханическое устройство, которое имеет в своем составе контактные группы, замыкающие или размыкающие электрические цепи. Тогда замыкание цепи приводит к тому, что по ней течет электрический ток, а в противном случае ток не течет.

С развитием физических наук и технологий реле были заменены на электронные устройства, которые на своих выходах могли формировать уровни либо высокого, либо низкого напряжения.

В обоих случаях одна из ситуаций могла обозначаться единицей, а другая нулем.

Булева алгебра оперирует высказываниями, каждое из которых может быть либо истинны - ***true*** (единица), либо ложным - ***false*** (ноль) и вводит аксиомы, теоремы и операции для их обработки и преобразований.

Алгебра Буля стала одним из разделов математики, названной впоследствии ***теорией дискретных устройств***.

Ее удобно использовать для описания законов работы цифровых дискретных устройств.

Булевой переменной ***x*** называется переменная, которая может принимать только одно из двух значений: 0 или 1. Этот факт закреплен в аксиоме

$$\begin{cases} \mathrm{x} = 1, \text{если } \mathrm{x} \neq 0 \\ \mathrm{x} = 0, \text{если } \mathrm{x} \neq 1 \end{cases}$$

т. е. никаких других значений переменная принимать не может.

Булеву переменную еще по-другому называют ***двоичной,*** или ***логической,*** переменной. Операции над этими переменными называют ***логическими***. Определены следующие операции над булевыми переменными:

1) ***логическое произведение,*** или ***конъюнкция*** двух переменных, обозначается

$$\mathrm{X}_1 \& X_2, \qquad \mathrm{X}_1 \wedge X_2, \qquad \mathrm{X}_1 * X_2, \qquad \mathrm{X}_1 X_2$$

и подчиняется следующим правилам (табл. 2.1):

Таблица 2.1

Таблица истинности операции логического произведения

X_1	X_2	X_1 & X_2
0	0	0
0	1	0
1	0	0
1	1	1

2) ***логическая сумма,*** или ***дизъюнкция*** двух переменных, обозначается

$$Y_1 \vee Y_2, \qquad \text{или} \quad Y_1 + Y_2$$

и подчиняется следующим правилам (табл. 2.2):

Таблица2.2

Таблица истинности операции логического сложения

Y_1	Y_2	$Y_1 + Y_2$
0	0	0
0	1	1
1	0	1
1	1	1

3) ***логическое отрицание,*** или ***инверсия*** переменной, обозначается горизонтальной чертой над переменной $\bar{A}$ (в некоторых пакетах прикладных программ инверсия переменной обозначается знаком апострофа или другим знаком около переменной), например,

$$\bar{a} = a' \qquad |b = \bar{b} \qquad đ$$

и подчиняется следующим правилам (табл.2.3):

Таблица 2.3

Таблица истинности операции логического отрицания

A	$\bar{A}$
0	1
1	0

Булевы переменные подчиняются следующим аксиомам:

0&0=0, 0&1=0, 1&1=1, 0+0=0, 1+1=1, 0+1=1,

причем они справедливы и для произвольного числа переменных:

0*0*0*…*0=0, 1+1+1+…+1=1.

Для операции инверсии справедливы следующее отношения:

$$\bar{\bar{a}} = a, \ \bar{\bar{\bar{b}}} = \bar{b}, \ \ \bar{0} = 1, \ \bar{1} = 0.$$

т. е. четное число инверсий дает саму переменную, а нечетное – инверсную. В булевой алгебре действуют следующие основные законы:

1) переместительный для дизъюнкции **a+b = b+a**,

для конъюнкции **ab = ba**;

2) сочетательный для дизъюнкции **a+(b+c)=(a+b)+c**,

для конъюнкции **a(bc)=(ab)c**;

3) распределительный для дизъюнкции **a+bc=(a+b)(a+c)**,

для конъюнкции **a(b+c)=ab+ac**.

Из аксиом и законов алгебры логики следует ряд важных теорем, свойств и правил, которые полезны при выполнении эквивалентных преобразований:

1) x+x+x+…+x = x, x x x…x = x;

2) a+1=1, abc+1=1

$$1 + x + ab + \overline{a}b\overline{d} + cef = 1$$

3) a&0=0

$$0\&abc\overline{d}\overline{aef\overline{d}} = 0 \quad 0 * a * b * \overline{c} * (\overline{c}d + abe) = 0$$

4) $\boldsymbol{a + \overline{a} = 1} \quad \boldsymbol{abc + \overline{abc} = 1} \quad \boldsymbol{a + a = a} \quad \boldsymbol{ab\overline{c} + ab\overline{c} = ab\overline{c}}$

$$a * \overline{a} = 0, \; abc * \overline{abc} = 0, \qquad a * a * a = a,$$

$$abc * abcd = abcd$$

5) законы склеивания

$$\boldsymbol{ab + a\overline{b} = a, \; a\overline{b}c\overline{d} + abc\overline{d} = ac\overline{d}}$$

$$(a + b)\left(a + \overline{b}\right) = a, \; \left(a + \overline{b} + \overline{c} + \overline{d}\right)\left(a + \overline{b} + c + \overline{d}\right) = a + \overline{b} + \overline{d})$$

5) законы поглощения

$$\boldsymbol{a + ab = a, \; a\overline{b}cd + ac = ac}$$

$$\boldsymbol{a(a + b) = a, \;\; ab\left(ab + a\overline{b}c + abcd + \overline{a}\overline{b}\overline{c}\right) = ab.}$$

7) теорема де Моргана: $\boldsymbol{\overline{a + b} = \overline{a}\overline{b}, \;\; \overline{ab} = \overline{a} + \overline{b}}$.

Примеры применения теоремы де Моргана:

$\overline{a\overline{bc}d} = \bar{a} + bc + \bar{d}, \;\; \overline{ab + b\overline{\overline{dc}}} = \overline{ab} * \overline{b\overline{\overline{dc}}} = (\bar{a} + \bar{b})(\bar{b} + dc)$.

2.2. Булевы функции. Способы их задания

Булевой функцией (*БФ*), или ***переключательной*** функцией (*ПФ*), или функцией ***алгебры логики*** (*ФАЛ*), от n переменных называется функция $f(x_1,x_2,x_3,\ldots,x_n)$, которая на любом наборе своих аргументов может принимать одно из двух значений: 0 или 1. Под набором аргументов понимается совокупность значений переменных, каждая из которых может быть равна 0 или 1. Для функции от n аргументов количество возможных наборов равно 2^n. На них может быть задано $(2^n)^n$ всевозможных БФ.

Способов задания БФ несколько. Одним из самых простых и наглядных является задание таблицей истинности.

Булевы функции f_1 и f_2 заданы таблицей 2.4. Таблицу следует понимать так, что в любой момент времени переменные abc могут принимать ***<u>любой из наборов</u>***, указанных в строках левой части таблицы. Например, abc=001, или abc=110, или abc=011 и т. д. В правой части таблицы указаны те значения функции, которые она должна принимать на данном наборе аргументов. То есть на наборе abc=001 f_1=0, f_2=0 при abc=111 f_1=1, f_2=0, а на набореabc=100 f_1=1, f_2=1 и т. д.

Таблица 2.4

Таблица истинности БФ f1 и f2

a	b	c	f_1	f_2
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

На рис. 2.1 таблица 2.4 развернута в виде диаграмм (осциллограмм) сигналов. На рисунке показаны изменения значений сигналов abc в дискретные моменты времени. Например, в момент T_0 переменные a=0, b=0, c=0, функции f_1=0, f_2=1. В момент T_3 a=0, b =1, c=1, функции f_1=0, f_2=0 и т. д.

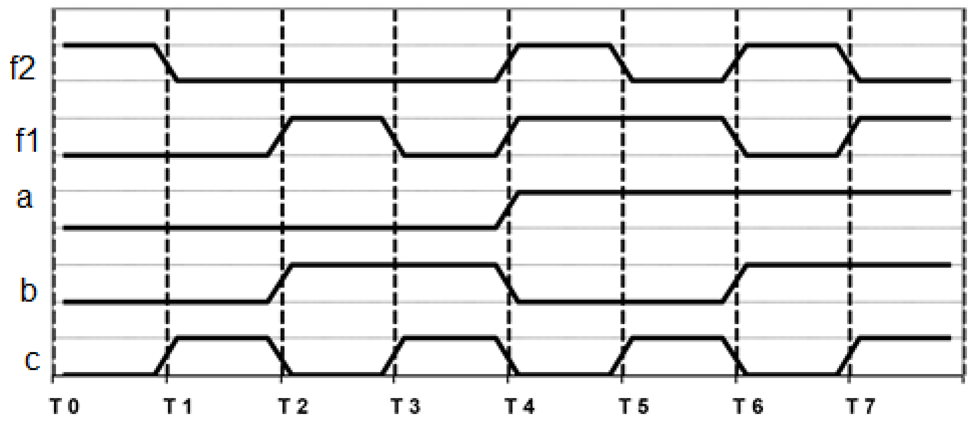

Рис. 2.1. Диаграмма сигналов таблицы 2.4

2.3 Совершенные формы булевых функций

Наборы логических переменных $x_1x_2\ldots x_n$, на которых БФ принимает единичные значения $f(x_1,x_2,\ldots,x_n)=1$, называют единичными наборами. Наборы логических переменных $x_1x_2\ldots x_n$, на которых БФ принимает нулевые значения $f(x_1,x_2,\ldots,x_n)=0$, называют нулевыми наборами.

Конъюнкцией называется логическое произведение произвольного числа аргументов. Конъюнкция называется ***элементарной***, если она содержит любое количество попарно различимых букв в прямой или инверсной форме. Например, конъюнкции являются элементарными

$$abc,\ \ \bar{a}\bar{b}c,$$

а вот конъюнкции

$$a\overline{bc},\ \ a(b+\bar{c})$$

не являются элементарными.

Назовем ***рангом*** R элементарной конъюнкции количество входящих в нее букв. Две конъюнкции назовем ***соседними***, если они зависят от одних и тех же аргументов, имеют одинаковый ранг и отличаются знаком инверсии только одного аргумента, например,

$$abc\bar{d} \text{ и } abcd,\ \ \bar{a}\bar{b}c\bar{d} \text{ и } \bar{a}\bar{b}cd.$$

Длиной функции L назовем сумму рангов всех входящих в функцию конъюнкций (т. е. количество букв в формуле).

Дизъюнктивной нормальной формой (ДНФ) БФ назовем дизъюнкцию (логическую сумму) конечного множества попарно различимых элементарных конъюнкций:

$$f_{ДНФ} = \mathrm{abc} + \mathrm{ac}\bar{d} + a\bar{b}\bar{c}d + \bar{a} \qquad (L = 11).$$

Конституентой единицы назовем элементарную конъюнкцию, содержащую ***<u>все аргументы</u>***, от которых зависит БФ. ***Совершенной дизъюнктивной нормальной формой*** БФ (**СДНФ**) назовем ДНФ, каждый член которой является конституентой единицы:

$$f = abcd + ab\bar{c}\bar{d} + adc\bar{d} + \bar{a}bcd + a\bar{b}c\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d} \qquad L = 24$$

Дизъюнкцией называется логическая сумма произвольного числа аргументов. Дизъюнкция называется ***элементарной***, если она содержит любое количество попарно различимых букв в прямой или инверсной форме. Например, дизъюнкции

$$a + b + c + d, \;\; \bar{a} + \bar{b} + d, \qquad a + c$$

является элементарными, а вот дизъюнкции

$$\boldsymbol{a + bc,\;\; ab + cd, \qquad a + \overline{c + d}}.$$

элементарными не являются.

Назовем ***рангом*** R элементарной дизъюнкции количество входящих в нее букв. Две дизъюнкции назовем ***соседними***, если они зависят от одних и тех же аргументов, имеют одинаковый ранг и отличаются знаком инверсии только одного аргумента. Например:

$$(a + b + c + d) \text{ и } (a + b + \bar{c} + d).$$

Длиной функции L назовем сумму рангов всех входящих в функцию дизъюнкций, т. е. количество букв в формуле. ***Конъюнктивной нормальной формой*** (КНФ) БФ назовем конъюнкцию (логическое произведение) конечного множества попарно различимых элементарных дизъюнкций:

$$f_{КНФ} = (a + \bar{b} + c)(a + \bar{b} + \bar{c} + d)(\bar{b} + \bar{c} + \bar{d}) \quad (L = 10).$$

Конституентой нуля назовем элементарную дизъюнкцию, содержащую ***все аргументы***, от которых зависит БФ. ***Совершенной конъюнктивной нормальной формой*** БФ (**СКНФ**) назовем КНФ, каждый член которой является конституентой нуля:

$$f_{СКНФ} = (a + b + c)(a + \bar{b} + c)(a + \bar{b} + \bar{c}) \qquad (L = 9)$$

2.4 Переход от табличного способа задания БФ к аналитическому

Пусть БФ f_1 и f_2 заданы табл. 2.5:

Таблица 2.5

Таблица истинности функций f_1,f_2

a	b	c	f1	f2
0	0	0	1	0
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

При переходе от табличного способа задания БФ к аналитическому в форме ДНФ можно получить только совершенные формы. Для получения СДНФ БФ записывается как логическая сумма конституент единицы. Для этого используется следующий алгоритм:

1) поочередно просматриваются все строки таблицы, отмеченные единичным значением функции;

2) конституента единицы записывается как логическое произведение переменных, причем, если переменная в наборе имеет значение 1, то она записывается в прямом виде, а если 0, то в инверсном, например СДНФ f1 будет иметь вид:

$$f_{1\text{СДНФ}} = \bar{a}\bar{b}\bar{c} + \bar{a}bc + a\bar{b}c + ab\bar{c} + abc \qquad L = 15$$

БФ f2 из табл. 2.4 представим в КНФ. Из таблицы можно получить только СКНФ. БФ записывается как логическое произведение конституент нуля. Для получения аналитического выражения БФ:

1) поочередно просматриваются все строки таблицы, отмеченные нулевым значением функции;

2) конституента нуля записывается как дизъюнкция переменных, причем, если переменная в наборе имеет значение 0, то она записывается в прямом виде, а если 1, то в инверсном, например f2 будет иметь вид:

$$f_{2\text{СКНФ}} = (a + b + c)(a + b + \bar{c})\left(a + \bar{b} + \bar{c}\right)\left(\bar{a} + \bar{b} + c\right)\left(\bar{a} + \bar{b} + \bar{c}\right) \; L = 15$$

Следует уяснить, что СДНФ и СКНФ — это просто разные **формы БФ**, и одна и та же БФ может быть записана в любой из них. Например, пусть БФ Z задана табл. 2.6.

Таблица 2.6

Таблица истинности функции Z

X1	X2	Z
0	0	0
0	1	1
1	0	1
1	1	0

Запишем эту функцию в обеих формах:

$$Z_{\text{СДНФ}} = \bar{a}\,\mathrm{b} + \mathrm{a}\bar{b},$$

$$Z_{\text{СКНФ}} = (a + b)(\bar{a} + \bar{b}).$$

Если в БФ, записанной в форме СКНФ раскрыть скобки и применить закон поглощения, то получится следующее:

$$Z = (a + b)(\bar{a} + \bar{b}) = a\bar{a} + a\bar{b} + b\bar{a} + b\bar{b} = a\bar{b} + \bar{a}b,$$

т. е. совпадает с СДНФ.

Если функция задана алгебраически, то может быть восстановлена таблица истинности. Например, задана функция P:

$$P_{ДНФ} = a\bar{b} + ab\bar{c}\,.$$

Вычисляются значения функции для всех наборов переменных:

$$P(0,0,0) = 0 \cdot \bar{0} + 0 \cdot 0 \cdot \bar{0} = 0 \cdot 1 + 0 \cdot 0 \cdot 1 = 0 + 0 = 0;$$
$$P(0,0,1) = 0 \cdot \bar{0} + 0 \cdot 0 \cdot \bar{1} = 0 + 0 = 0;$$
$$P(0,1,0) = 0 \cdot \bar{1} + 0 \cdot 1 \cdot \bar{0} = 0 + 0 = 0;$$
$$P(0,1,1) = 0 \cdot \bar{1} + 0 \cdot 1 \cdot \bar{1} = 0 + 0 = 0;$$
$$P(1,0,0) = 1 \cdot \bar{0} + 1 \cdot 0 \cdot \bar{0} = 1 + 0 = 1;$$
$$P(1,0,1) = 1 \cdot \bar{0} + 1 \cdot 0 \cdot \bar{1} = 1 + 0 = 1;$$
$$P(1,1,0) = 1 \cdot \bar{1} + 1 \cdot 1 \cdot \bar{0} = 0 + 1 = 1;$$
$$P(1,1,1) = 1 \cdot \bar{1} + 1 \cdot 1 \cdot \bar{1} = 0 + 0 = 0.$$

Вычисленные значения функции заносятся в таблицу (табл. 2.7).

Таблица 2.7

a	b	c	P
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

По табл. 2.7 можно записать СДНФ:

$$P_{СДНФ}(a, b, c) = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}.$$

Для функции R, заданной в конъюнктивной форме

$$R_{КНФ} = \bar{a}(a + \bar{b})$$

вычислены значения функции, которые занесены в табл. 2.8:

$$R(0,0) = \bar{0} \cdot (0 + \bar{0}) = 1 \cdot (0 + 1) = 1 \cdot 1 = 1;$$
$$R(0,1) = \bar{0} \cdot (0 + \bar{1}) = 1 \cdot 0 = 0;$$
$$R(1,0) = \bar{1} \cdot (1 + \bar{0}) = 0 \cdot 1 = 0;$$
$$R(1,1) = \bar{1} \cdot (1 + \bar{1}) = 0 \cdot 1 = 0.$$

Таблица 2.8

a	b	R
0	0	1
0	1	0
1	0	0
1	1	0

$$R_{СКНФ} = (a + \bar{b})(\bar{a} + b)(\bar{a} + \bar{b}) .$$

2.4 Числовой способ задания БФ

Существует еще один способ задания БФ. Он основан на следующем: если каждой логической переменной поставить в соответствие **разряд** двоичного числа, то каждому **набору** булевых переменных будет соответствовать **двоичное** число, а ему десятичное.

Тогда БФ можно задать перечислением наборов, на которых она принимает только значения равные единице, или только значения равные нулю.

Так БФ f1 из табл. 2.5 в форме СДНФ может быть записана:

$$F1_{СДНФ}=\Sigma(0,3,5,6,7),$$

а БФ f2 в форме СКНФ: $F2_{СКНФ}=P(0,1,3,6,7)$.

В записи формул используются знаки Σ для ДНФ (сумма) и P (произведение) для КНФ.

От числового способа задания можно легко перейти к табличному способу задания и к записи БФ в виде формулы.

2.5 Применение законов склеивания для минимизации булевых функций

Две БФ называются **эквивалентными**, если они принимают одинаковые значения на одних и тех же наборах аргументов. Одна и та же БФ может быть записана разными формулами. Из нескольких формул предпочтительней та, которая имеет наименьшую длину. Для получения более короткой формулы применяются эквивалентные преобразования с использованием правил склеивания и поглощения. Процедура получения минимальной формулы БФ называется ***минимизацией***. Алгоритм минимизации БФ, заданной в форме СДНФ, следующий:

1) склеиванию подлежат только соседние конституенты единицы. В результате склеивания образуется конъюнкция, называемая импликантой. Импликанта, полученная склеиванием двух конъюнкций, ранг которых равен R, имеет ранг R-1;

2) одна и та же конституента единицы может принимать участие в нескольких операциях склеивания;

3) к полученным импликантам опять применяется операция склеивания (до тех пор, пока это возможно);

4) БФ записывается, как дизъюнкция полученных импликант.

Исходя из этого для БФ f1 из табл. 2.5

$$f_{1СДНФ}=\bar{a}\bar{b}\bar{c}+\bar{a}bc+a\bar{b}c+ab\bar{c}+abc \qquad (L=15)$$

1 2 3 4 5

- первая конъюнкция не склеивается ни с одной другой;
- вторая склеивается с пятой $\bar{a}bc + abc = bc$,
- третья – с пятой $a\bar{b}c + abc = ac$,
- четвертая – с пятой $ab\bar{c} + abc = ab$,
- полученные конъюнкции больше не могу быть склеены. БФ f1 запишется как:

$$f_1 = \bar{a}\bar{b}\bar{c} + bc + ac + ab \qquad (L = 9),$$

т. е. очевидно, что полученная ДНФ короче, чем исходная СДНФ. Алгоритм минимизации БФ, заданной в форме СКНФ:

1) склеиванию подлежат только соседние конституенты нуля. В результате склеивания образуется дизъюнкция, которая имеет ранг R-1;

2) одна и та же дизъюнкция может принимать участие в нескольких операциях склеивания;

3) к полученным дизъюнкциям опять применяется операция склеивания (до тех пор, пока это возможно);

4) БФ записывается, как произведение полученных дизъюнкций.

Минимизация БФ f2 из табл. 2.5

$$f_{2СКНФ} = (a + b + c)(a + b + \bar{c})\left(a + \bar{b} + \bar{c}\right)\left(\bar{a} + \bar{b} + c\right)\left(\bar{a} + \bar{b} + \bar{c}\right)$$

1 2 3 4 5

- первая дизъюнкция склеивается со второй $(a + b + c)(a + b + \bar{c}) = a + b$,
- вторая склеивается с третьей $(a + b + \bar{c})\left(a + \bar{b} + \bar{c}\right) = a + \bar{c}$,
- четвертая – с пятой $\left(\bar{a} + \bar{b} + c\right)\left(\bar{a} + \bar{b} + \bar{c}\right) = \bar{a} + \bar{b}$,
- к полученным дизъюнкциям операция склеивания уже не применима, поэтому БФ f2:

$$f_2 = (a + b)(a + \bar{c})\left(\bar{a} + \bar{b}\right) \quad (L = 6),$$

т. е. очевидно, что полученная КНФ короче, чем исходная СКНФ. Для f2 операции склеивания можно выполнить и по-другому:

- первую – со второй, четвертую – с пятой, как и в первом случае;
- а вот третью можно склеить не со второй, а с пятой
- полученные дизъюнкции не могу быть склеены, поэтому можно записать:

$$f_2 = (a + b)(a + \bar{c})\left(\bar{b} + \bar{c}\right) \quad (L = 6).$$

Оба варианта функции f2 имеют одинаковую длину L=6, поэтому они равноценны.

2.6 Применение законов поглощения для минимизации булевых функций

Применение законов поглощения также позволяет минимизировать БФ, заданных не в совершенной форме. Пусть БФ задана в форме ДНФ:

$$f = x_1x_2 + x_1\bar{x}_2 + x_1\bar{x}_3\bar{x}_5x_6 + x_1x_3x_4\bar{x}_5 + \bar{x}_1\bar{x}_3\bar{x}_4.$$

Применим закон склеивания к первой и второй конъюнкции:

$$f = x_1x_2 + x_1\bar{x}_2 + x_1\bar{x}_3\bar{x}_5x_6 + x_1x_3x_4\bar{x}_5 + \bar{x}_1\bar{x}_3\bar{x}_4 =$$

$$= x_1 + x_1\bar{x}_3\bar{x}_5x_6 + x_1x_3x_4\bar{x}_5 + \bar{x}_1\bar{x}_3\bar{x}_4.$$

В соответствии с законом поглощения (переменная x_1 поглощает все конъюнкции, содержащие x_1):

$$f = x_1 + x_1\bar{x}_3\bar{x}_5x_6 + x_1x_3x_4\bar{x}_5 + \bar{x}_1\bar{x}_3\bar{x}_4 = x_1 + \bar{x}_1\bar{x}_3\bar{x}_4.$$

Для функции P:

$$P = (x_2 + \bar{x}_3)(x_2 + x_3)(x_2 + \bar{x}_1x_4)(x_2 + \bar{x}_3\bar{x}_4)(\bar{x}_1 + \bar{x}_2\bar{x}_4) =$$

$$= x_2(x_2 + \bar{x}_1x_4)(x_2 + \bar{x}_3\bar{x}_4)(\bar{x}_1 + \bar{x}_2\bar{x}_4) = x_2(\bar{x}_1 + \bar{x}_2\bar{x}_4) = \bar{x}_1x_2.$$